

9. Prateek

Program Name: Prateek.java

Input File: prateek.dat

The Uncannily Intelligent Lexicographers (UIL) is testing new methods of compressing dictionaries. A dictionary is a collection of words. The words in the dictionary can be stored in any order, and different ways to store a word in the dictionary can take up differing amounts of space. The UIL's dictionary structure has two methods of representing a word:

1. Record a whole word. By UIL mandate, this takes A bytes of memory, regardless of the length of the word.
2. Record a word by difference. This method begins with a B byte reference to a previous word T and D bytes for each difference between T and the current word S . The total space is $B + D * \text{dist}(S, T)$ bytes of memory where $\text{dist}(S, T)$ is the Levenshtein distance between S and T .

The Levenshtein distance between two words is the number of single-character additions, deletions, and changes to go from one word to the other. For example, the Levenshtein distance between "cat" and "dog" is 3 (change each character). The Levenshtein distance between "apple" and "orange" is 5 (delete the 'l', change the first 'p' to an 'n', change the second 'p' to a 'g', and add 'o' and 'r' at the beginning).

Given a dictionary of words, what is the minimum number of bytes needed to encode all the words?

Input: The first line of input is T ($1 \leq T \leq 30$), the number of test cases. Each test case begins with 4 integers $N A B D$ where N ($1 \leq N \leq 100$) is the number of words in the dictionary, A is the number of bytes storing a whole word takes, B is the number of bytes to record a word by difference, and D is the cost per difference. $1 \leq A, B, D \leq 1,000$. Then follow N lines, each containing a word in the dictionary. Each word consists of at most 20 lowercase letters (a-z).

Output: For each test case, output the minimum cost to send all words, formatted with the case number as in the samples.

Sample input:

```
2
3 10 4 1
apple
orange
strawberry
5 4 1 1
ab
ac
xy
yz
az
```

Sample output:

```
Case #1: 29
Case #2: 13
```

Sample Explanation: For the first sample, the best strategy is to send "apple" and "strawberry" normally, and then send "orange" as a difference of "apple". This saves one byte over sending all 3 words as whole words.